



 eBook

オブザーバビリティ (可観測性) 入門

可観測性でひろがるオープンで
プログラマブルな未来

目次

INTRODUCTION	03
チャプター 1：新しいアーキテクチャには新しいモニタリングが必要	04
チャプター 2：可観測性の時代	06
チャプター 3：可観測性の 3 つのポイント	07
チャプター 4：オープンインストゥルメンテーション	08
チャプター 5：関連付けデータとキュレートデータ	10
チャプター 6：プログラマビリティ	13
チャプター 7：あらゆる機能を一元化	14

Introduction

New Relicの創業者、ルー・サーニーがAPM（アプリケーションパフォーマンスモニタリング）を開発した時、データセンターで稼働するモノリシックなアプリケーションのコードを詳細に可視化したことが革新的な点でした。その後、開発と運用に携わるエンジニアが活用できるよう、ルー・サーニーはこれをSaaSソリューションとして公開しました。クラウド、マイクロサービス、コンテナ、サーバーレス、DevOps、SRE（サイトリライアビリティエンジニアリング）などの新しいテクノロジーや取り組みによって開発速度が向上し、ソフトウェアの本番環境へのデプロイがスムーズになる一方で、複雑性も増しています。

New RelicはAPMのパイオニア企業として、この最新のソフトウェアチームが直面する課題を解決するには、新しいアプローチが必要だと考えています。複雑さを緩和し、エンジニアがいつでも最新のシステムを使える状態にするとともに、優れた顧客体験を実現するために必要なのは「Observability（オブザーバビリティ／可観測性）」です。

「可観測性」によってエンジニアはソフトウェアを通じた優れた顧客体験の提供できるようになるため、最先端のデジタル企業を中心に可観測性への注目が高まっています。

しかし、ここで1つ注意しておきたいのは、可観測性と監視（モニタリング）は同義ではないということです。

監視によって、ソフトウェアチームはシステムに関するデータを収集し、エラーや問題の発生時に迅速に対応できるようになります。つまり、監視とはデータを収集することを目的としてシステムを構築することであり、問題を検知し、迅速に対応する体制を整えることが目的です。

一方、可観測性とはエラーや問題が発生したことを検知するだけでなく、それが「なぜ」起きたのか、その原因を探り、改善のアクションに繋げることを目的としてデータを収集します。最新のデジタルサービス、ソフトウェアに必要なのは、緊急事態が起きても迅速な対応と原因の特定、解決を可能にする後者の「可観測性」です。

可観測性は、最新のソフトウェアチームにさまざまなメリットを提供します：

- 高品質なソフトウェアの提供
- 継続的なイノベーションの実現
- クラウドや最新ツールへの投資の最適化
- デジタルビジネスのパフォーマンスをリアルタイムに可視化

New Relicでは、可観測性に不可欠なデータのタイプは、メトリクス、イベント、ログ、トレース（M.E.L.T.）だと考えています。しかし、可観測性には単なるデータ以上の意味があります。

システムの可観測性を確立するにはどうすれば良いでしょうか？また、可観測性によってどんな効果が期待できるのでしょうか？ New Relicが考える、最新のデジタル企業が抱える4つの課題と、その解決策である可観測性の実現に必要な3つのポイント、「オープンインストゥルメンテーション」、「関連付けデータ」、「プログラマビリティ」を導入する必要があります。このeBookでは、こうした最新のトレンド、課題、ポイントをご紹介します。

CHAPTER 1: 新しいアーキテクチャには新しいモニタリングが必要



この5年、10年での技術革新のスピードは驚異的であり、さまざまなソフトウェアチームに大きな影響を及ぼしてきました：

- **迅速なイノベーションへのプレッシャー**：ソフトウェアチームは、新しい機能や UX を競合他社よりも早く、しかも頻繁にマーケットに投入しなければならないという大きなプレッシャーに直面しています。クラウドの出現によって参入障壁が下がり、競争力は高まると同時に、ソフトウェアチームにはこれまで以上に迅速な開発を求められるようになりましたが、そのためのリソースは減っているのが実状です。パフォーマンスの高い企業（High Performer）は、1時間に1回から1日に1回の頻度でソフトウェアをデプロイし、パフォーマンスの優れた企業（Elite Performer）は、必要に応じて1日に複数回のデプロイを行います。
- **顧客からの高い期待値**：顧客の企業への期待値は上がり続けていますが、逆に寛容さ、許容度は下がっています。サービスが遅い、エラーが発生しやすい、設計が不十分など、顧客体験の質が悪い場合、成功は期待できません。期待した結果が得られなければ、顧客が戻ってくることは二度とありません。モバイルアプリ開発者の Dot Com Infoway によると、モバイルでクラッシュやフリーズ、エラーが発生した場合、[62%の人がアプリをアンインストールする](#)ということです。ソフトウェアの提供において優れたパフォーマンスの企業は、ユーザーに影響を与えるインシデントや不具合が発生した場合、1時間以内にサービスを復旧するのに対し、パフォーマンスの低い企業の場合は、サービスの復旧に1週間から1ヵ月の時間を要することがあります。¹

- **豊富なテクノロジーの選択肢**：現在の組織は、数多くのクラウドサービス上にマイクロサービスアーキテクチャと分散システムを構築しています。このようなサービスによって導入と利用が容易になり、シームレスな連携が可能になります。さまざまなシステムやサービスを選択し、テクノロジースタックのサポートに必要なものが揃っているため、管理工数も削減することができます。
- **DevOpsと自動化の発展**：企業は、自社で運用するサービスの設計、開発、運用を担当するチームを中心に組織化されています。社内のプラットフォームチームがサービスとして提供する共通のツールを活用することもあります。自動化によって、反復性があり、価値の低い作業（あるいは負荷の高い労働）が削減され、信頼性が向上します。クラウドネイティブアーキテクチャでは、スタック内のすべてがソフトウェアによって制御され、表面積全体がプログラム可能です。しかし自動化もソフトウェアによって行われるため、失敗することもあります。チームは顧客に直接サービスを提供するアプリケーションと同様、CI/CDなどの自動化ツールを正確に監視する必要があります。可観測性で重要なことは、システム内のすべてのコンポーネントに関するデータを収集することです。

こうした傾向から、最新のシステムが抱える4つの課題が生まれています。

1. **複雑化**：クラウドネイティブテクノロジーは、アプリケーションの設計、開発、運用の方法を一変させましたが、その一方で、アプリケーションの保守を担当するチームにとっては複雑度が増しています。モノリシックなアプリケーションがマイクロサービスにリファクタリングされ、コンテナのライフタイムを数分以下で測定できるようになると、ソフトウェアチームはコンスタントに変化するサービスを使用できるようになります。個々のアプリケーションは数十個のマイクロサービスに分解される可能性があるため、運用チームは複雑性の問題に直面することになります。
2. **ハイリスク**：頻繁なデプロイとダイナミックなインフラストラクチャは、リスクが入り込む回数も増えます。リスクが増えたことで、デプロイ頻度の低い時代よりも、リスクの迅速な検出とロールバックの重要性が高まっています。ソフトウェアのリリースまでの時間を短縮するため、アジャイルなプラクティスと継続的な提供プロセスを採用する企業が増えており、監視と保守が必要なソフトウェア領域がさらに増えることになります。
3. **スキルの格差**：マイクロサービスアーキテクチャが爆発的に増加することで、新たな課題が生まれ、ソフトウェアチームはアプリケーションの設計、構築、導入方法を再考しなければならない状態です。チームメンバーは、あまり知識のないアプリケーションについても理解し、トラブルシューティングすることが求められます。たとえば、データベースのエキスパートは、ネットワークやAPIについても知っておく必要があります。最大の問題は、チームとして使い方を学ばなくてはならない新しいテクノロジーもテクノロジーの種類も多すぎて、1人でマスターするには限度があるということです。つまり、自分たちの作業に当てはめて、こうしたテクノロジーを理解する方法が必要なのです。
4. **ツールの煩雑化**：ハイブリッド環境、本番環境での数千個のコンテナ、1日に複数回のデプロイによって、膨大な量のテレメトリデータが生成されます。複数の監視ツールと、最も重要なデータの検出と関連づけ、または問題の検出と解決に必要なコンテキストの切り替えを同時に実行した場合、生産環境で問題が発生したとしても、チームにとって貴重な時間を費やさなければならなくなります。

こうした傾向や課題だけでなく、技術の進化のスピードを考えると、チームにとって必要なのは、複雑な手間とリスクを減らし、経費を低く抑えるシングルソリューションです。スキルの格差を埋め、目的に合わせて簡単に使え、わかりやすく、ナビゲートも容易なソリューションであることです。また、組織内のどのチームも可観測性データを一カ所で表示でき、速やかに意味を導き出して、適切な対応が取れることも重要です。

1. 「加速化：2019年DevOpsの状況」DORA（2019年9月）

チャプター 2：可観測性の時代



一般的な監視（モニタリング）は、少なくとも Unix 初期の時代までさかのぼりますが（最初のエディションは1971年に発表されました）、アプリケーションパフォーマンスモニタリング（APM）という単語が広く使われるようになったのは、2000年代初頭のことでした。それ以降、クラウドを含むテクノロジースタック全体のパフォーマンスとユーザーエクスペリエンスの具体的なメトリクス、トレース、アラートを示すため、監視（モニタリング）は進化を遂げてきました。

環境がどんどん複雑化し、ソフトウェアチームと組織が将来的に成功を収めるには、非常に重要なのが可観測性です。可観測性の実現により、チームはすべてのパフォーマンスデータを一カ所にまとめてダッシュボード化し、リアルタイムに確認することができるので、問題を正確かつ迅速に特定し、原因を把握することができます。その結果、優れた顧客体験を提供することができます。

可観測性は、まったく新しいコンセプトではありません。本来は工学と制御理論から生まれたもので、ハンガリー系アメリカ人のエンジニア、Rudolf E. Kálmán氏によって線形動的システムに導入されました。工学と制御理論に適用される可観測性の一般的な定義は、システムの内部状態を外部出力の知識から推測するための手段だと考えられています。

ソフトウェアのライフサイクルにおいて、「可観測性」とは、システムの動作を全体的に理解するために、メトリクス、イベント、ログ、トレースを収集、視覚化、分析することを意味します。可観測性は、問題が発生した原因を伝えるもので、何らかの問題があるという事実のみを通知する監視とは異なります。

Uber Technologiesのライター兼ソフトウェアエンジニアのYuri Shkuro氏は、この違いを次のように説明しています。「監視は事前に決めておいた項目を測定するもので、可観測性とはシステムに関して知らないことを尋ねることができる機能です。」

可観測性に不可欠なデータタイプは、メトリクス、イベント、ログ、トレースであり、イベントは可観測性ソリューションの一部に組み込むべき重要な（そして見落とされがちな）テレメトリタイプです。これについては、後ほどお話します。つまり、あらゆるデータを測定し、テレメトリデータを用いてシステム内の関係性や依存関係だけでなく、性能や健全性に関する基本的な知識を形成することで、可観測性を実施しているのです。しかし、New Relicは可観測性の実現には3つのポイントがあると考えています。

CHAPTER 3：可観測性の3つのポイント



ここまで、可観測性とはエラーや問題が発生した事実だけでなく、その原因を示す実用的なデータを収集するため、ツールでシステムを計測することとして定義しました。こうした原因を特定する能力があるからこそ、チームは根本原因を解決し、システムの信頼性を確保することができるのです。New Relicでは、システムの可観測性を実現するには、**3つのコア要素**が必要だと考えています。

- 1. オープンインストゥルメンテーション：**New Relicではオープンインストゥルメンテーションをアプリケーション、サービス、インフラストラクチャ、コンテナ、クラウドサービス、サーバーレス機能、モバイルアプリケーション、またはデータを送信するエンティティからの、オープンソースまたはベンダー固有のテレメトリデータのコレクションと定義しています。つまり、ビジネスクリティカルなアプリケーションとインフラストラクチャの全体を可視化することを示します。
- 2. 関連付けエンティティ：**テレメトリデータを生成するエンティティを識別して接続するには、すべてのテレメトリデータを分析する必要があり、エンティティとデータ間の相関を生成するためにはメタデータを組み込む必要があります。この2つのアクションは、大量のデータからコンテキスト（文脈）と意味を引き出します。このコンテキストから、キュレーションはシステムやテクノロジーの視覚的なモデルとして、追加の設定なしに提供することができます。エンティティ連携の利点は、人工知能を応用してさらに多くの意味を導き出せることです。Applied Intelligenceとは、機械学習やデータサイエンスを応用してデータのパターンや異常を探し、人間の意思決定や行動を促すことです。

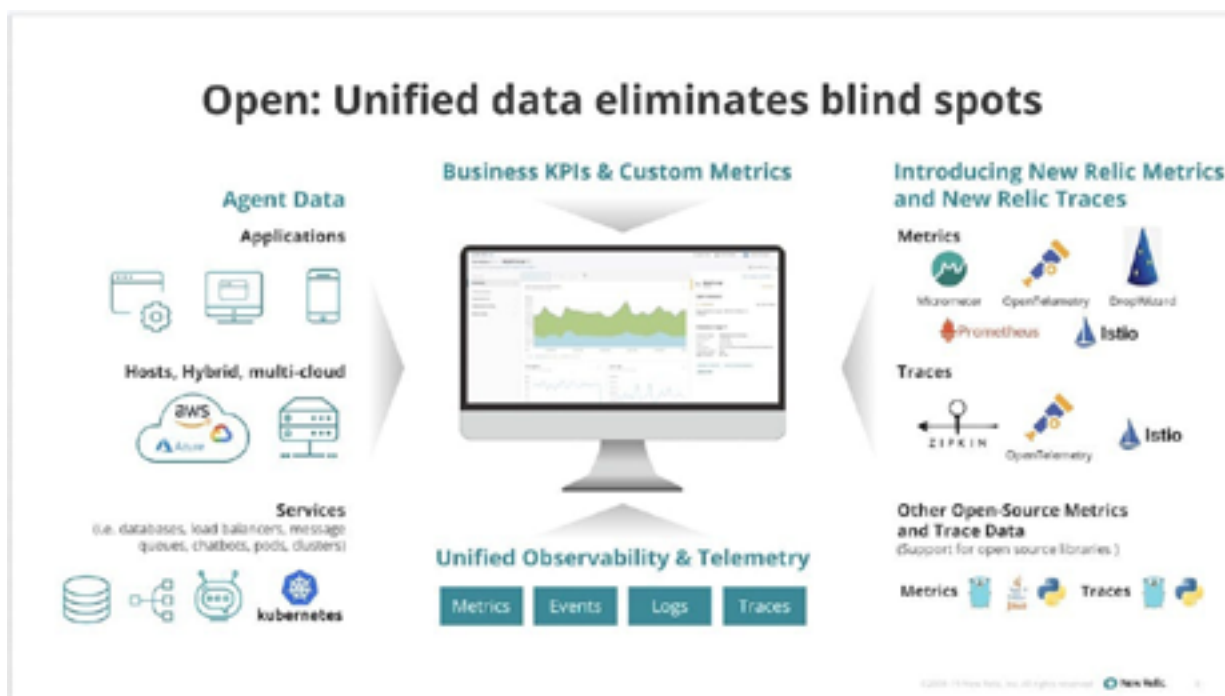
3. **プログラマビリティ**: 企業のビジネスはそれぞれ異なるため、どのような自動キュレーションでも、ビジネスの様々なニーズを満たしたり、すべてのユースケースに適合したりすることはできません。ビジネスには、すべてのテレメトリデータの上に、重要なビジネスデータとディメンジョンを組み合わせ、独自のコンテキストとキュレーションを作成する必要があります。こうしたニーズに対して、すべてのテレメトリデータを基にアプリケーションを構築する機能を New Relic は提供しており、それがユニークな価値でもあります。たとえば、ビジネスプロセスにおけるエラーや失敗のコストを明確に表示し、それらの失敗を集計したコスト、その原因を見つけるためのドリルダウンできるデータを提供する能力を持っています。

最新のソフトウェアをサポートするため、可観測性がどのような進化を続けているかを説明します。「[可観測性の10の原則: 最新のソフトウェアで成功するための道標](#)」

チャプター 4：オープンインストゥルメンテーション

2008年に New Relic を立ち上げた際、可観測性のテレメトリを収集する最善の方法はエージェントを使用することでした。ソフトウェアの開発者や運用チームは、アプリケーションやホストの内部にエージェントを配置し、こうしたエージェントでメトリクス、イベント、トレース、ログデータを収集し、独自の方法でパッケージ化し、送信して集計や表示を行っていました。

テレメトリデータを収集するには効率的なルートであることに変わりはありませんが、業界は変化しています。現在、テレメトリのソース（情報源）はさらに増えています。ソフトウェア開発向けのオープンシステムやフレームワークには、メトリクス、イベント、ログ、トレースが組み込まれており、共通のフォーマットで出力されます。可観測性のためには、オープンソースと独自ソースの両方からデータを収集し、それを一つの場所にまとめる必要があります。追加すべきだと判断できるところには自動でインストゥルメンテーションを適用し、最も可視性が必要なところにはインストゥルメンテーションを追加する必要があります。



M.E.L.T.：簡単な定義

ほとんどの場合、可観測性は**メトリクス**から始まります。コストを抑えて収集、保存できます。また、迅速な分析ができる規模で、全体的な健全性を評価するのに最適な方法です。このため、Prometheus、Telegraf、StatsD、DropWizard、Micrometerなど、メトリクス収集用のツールが数多く登場しています。Elasticsearchなどの時系列に応じたオープンなデータストアだけでなく、独自のフォーマットを構築してメトリクスを収集する企業も少なくありません。可観測性の実現には、最新のデジタル企業でさまざまなチームが導入しているソースからメトリクスを利用できるようにする必要があります。

トレースは、分散アーキテクチャにおける個々のコール（呼び出し）のエンドツーエンドレイテンシーを示すのに便利です。これらのコールは、システムによるさまざまなカスタマージャーニー（購買までのプロセス）に関して独自のインサイトをもたらします。トレースを使用することで、エンジニアはこれらのプロセスを理解し、ボトルネックを検出すると同時にエラーを特定し、修正と最適化を行うことができます。メトリクスと同様、高度な組織が作ったカスタムソリューションから多くのツール（Jaeger、Zipkin、AWS X-rayなど）が登場しています。

W3C Trace Contextは、プロセスの境界線を超えて「トレースコンテキスト」を普及させるための標準になるでしょう。トレースコンテキストは、複雑な分散システムをまたいで発信元の呼び出し（親スパンとその子スパン）を追跡し、システムを介したデータの流れを追跡するための標準的な方法を提供します。開発者がトレースコンテキストに標準を使用すると、多くの異なるシステムからのスパンを信頼性の高い形でつなぎ直して、オブザーバビリティ・プラットフォームでの表示や検索が可能になります。また、トレースコンテキストには重要なタグやその他のメタデータが含まれているため、検索や相関関係をより強力にすることができます。

Cloud Native Computing Foundation (CNCF)の一部である**OpenTelemetryプロジェクト**は、メトリクスとトレースコレクションをオープンフォーマットで統合します。OpenTelemetryを採用する組織が増えれば、実行時にバイトコードインストゥルメンテーション用のエージェントを実行する必要性を減らすような標準化がなされると予想されます。CNCFのKubernetesやIstioのようなツールの幅の広さとその急速な普及を考えると、OpenTelemetryはテレメトリのソースとして、現代のソフトウェアの中でユビキタスなものになると考えられます。

ログは、エンジニアが問題の根本原因を追求する「ディープ」なデバッグモードに入った場合、特に重要です。ログは、忠実度の高いデータとイベントの詳細なコンテキストを提供するため、エンジニアはミリ秒単位で起こったことを再現することができます。メトリクスやトレースと同様、ログの収集、フィルタリング、エクスポートの手間を軽減するツールが登場しています。Fluentd、Fluent Bit、Logstash、AWS CloudWatchなどの一般的なソリューションのほか、新しい標準も数多く現れています。

こうしたメトリクス、ログ、トレースのためのこれらのプロジェクトはすべて、誰もが容易にインストゥルメンテーションを活用できる未来に向けて構築されています。

イベントは、あらゆる可観測性ソリューションの一部でなければならない重要な（そして見落とされがちな）テレメトリタイプのデータです。しかし残念なことに、イベントとログはいくつかの類似点を共有していますが、この2つはしばしば誤って混同されます。イベントは、分析の重要なポイントを個別かつ詳細に記録したものです。しかし、ログが提供する詳細レベルよりも高い抽象度を含んでいます。ログはシステム内で起こったすべてのことを包括的かつ離散的に記録したものであり、イベントは起こった重要なことを選択して記録したものであり、その記録にメタデータを付加して文脈をより鮮明にします。例えば、New Relicがトランザクション・イベント、つまり、トランザクション・イベントの個々のインスタンスを収集している場合、イベントは、その記録にメタデータを付加して、コンテキストを鮮明にします。

イベントとは？

イベントとは、可観測性で最も重要なデータのタイプです。イベントはログとは明確な違いがあります。イベントは、重要な分析ポイントを個別かつ詳細に記録したのですが、ログの情報よりも抽象度の高いデータを含むのが特徴です。アラートはイベントに該当します。デプロイメントもイベントで、トランザクションもエラーもそうです。イベントは、リアルタイムで詳細な分析を実行することができます。

基本的なインストゥルメンテーションを提供するほとんどのオープンソースツールには、データを収集、保管し、分析に利用できるようにするための独立したデータストアが付属していますが、これは可観測性の実用性を損ねることになります。エンジニアやチームが複数のツールを把握し、理解しなければならなくなるからです。統合されたデータストアがないと、問題（最悪の場合は緊急事態）が発生した場合、エンジニアは複数のツールを使用してコンテキストを切り替え、問題の原因を特定する必要があります。オープンな可観測性ソリューションの場合、ソースに関係なく、すべてのデータの相互運用性があります。エンティティとその間の接続を自動的に作成し、重要なコンテキストを提供します。

チャプター 5：関連付けデータとキュレートデータ

任意の場所からテレメトリデータを取得して、一元的に集めるのはスタートとしては良いですが、十分とは言えません。データは、エンティティ間の関係を理解できるように接続されている必要があり、ビジネスとの関係を理解できるようにメタデータと関連付けられている必要があります。こうした関連付けにより、データにコンテキストと意味が付加されます。たとえば、コンテキストは、データに関する最も重要な情報を表示し、その特定の環境をモデル化するキュレーションされたビューにつながります。さらに、すべてのテレメトリデータと接続が一カ所に保存されている場合、非常に大きなデータセットにインテリジェンスを適用して、ダッシュボードを見ている人間には容易に識別できないパターン、異常、相関関係を表面化させることができます。

本質的には、システム内のすべてのエンティティが、ある時点でどのように相互に関連しているかを確認する方法が必要です。システムが日ごと、時間ごと、分ごとに変化するときに、システムのメンタルマップを維持することは、単純に実現不可能です。また、これらの関係を管理するためにコンフィグレーションに頼ることも現実的ではありません。チームが新しいサービスを追加したり、古いサービスをリファクタリングしたり、一時的なアプリケーション・インスタンスをスピンアップしたりシャットダウンしたりすると、メンタル・マップを維持することは不可能になります。しかし、エンティティ、その接続、および関係性は、観察可能性のための本質的なコンテキストの一部です。

コンテキストはメタデータとディメンジョンがなければ成立しません。システムやビジネス、あるいはアプリケーションによっては、貴重なデータの範囲が膨大になる可能性があります。たとえば、eコマースアプリケーションの場合、有用なコンテキストには以下が含まれますが、これに限定されません：

- アプリケーション、ランブック、およびコードリポジトリを所有するチームの詳細情報
- Docker または展開されているクラウドプロバイダーからのタグ
- サービスのタイプと機能
- 展開されているリージョン
- アップストリームとダウンストリームの依存関係

- 展開または変更イベント
- アラートのステータス
- 実行するトランザクションに関連するすべてのトレースまたはログデータ
- 付加的なビジネス情報（例：カートの価値）

データ可視化のキュレーションは、接続され定義されたエンティティを表面化するための強力なツールです。コンテナ内で実行されている Java アプリケーション・プロセス、SQS からの呼び出し後に DynamoDB を呼び出す AWS Lambda、動的なデプロイメントを実行している Kubernetes クラスタなどを表現するための最適な方法はすでに知っています。また、多忙な SRE や DevOps エンジニアにとって、これらの環境をダッシュボードでモデリングするために時間を要するのは貴重な時間の無駄遣いになってしまいます。可観測性プラットフォームは、業界リーダーのベストプラクティスを取り入れ、健全性の最も重要なシグナルを明らかにするだけでなく、エンジニアが問題を迅速にトラブルシューティングできるようなインタラクティブな体験を提供する必要があります。特定のユビキタス技術のための可視化やダッシュボードを手動で作成するのは、簡単な作業です。

コンテキストを通じたキュレーションは、複雑なデジタル企業におけるスキルギャップ（能力の格差）という課題の解消にも有効です。組織内のすべての人が、複雑なシステム内のフローと依存関係を視覚化し、環境全体に関連するすべての情報を確認することができます。このキュレーションはさまざまなシステムをうまくモデル化しているので、特定の技術やコードに関して知識がない人でも理解しやすくなっています。

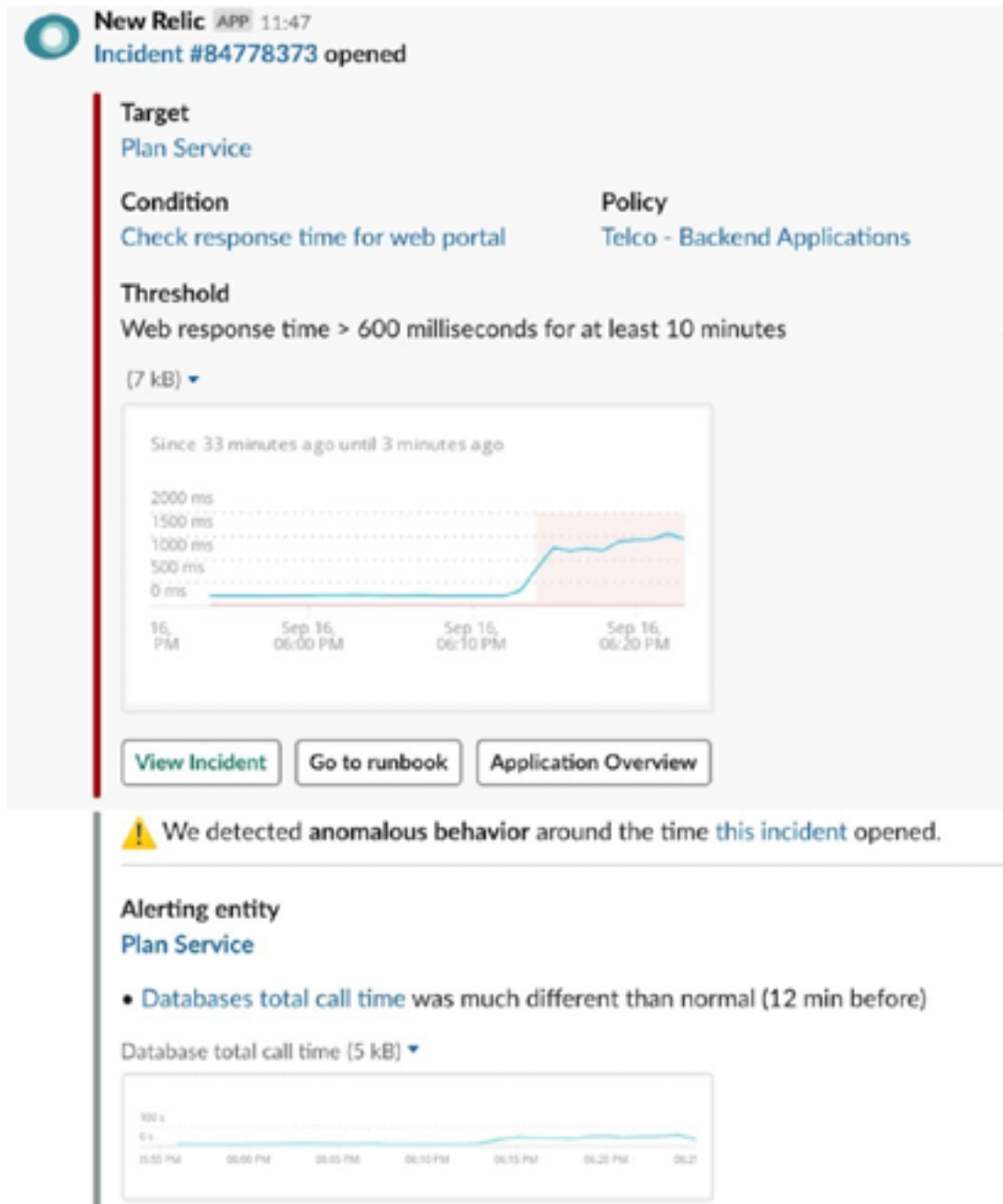
システムが正常に動作していない状況で迅速に対処できなければ、可観測性には何の意味もありません。Applied Intelligence は、機械学習と予測分析を通じて、可観測性データを取得し、有意義で実用的なデータにします。応用知能は、IT 運用では人工知能、または産業分析企業の Gartner 社では AIOps と呼ばれ、適切な行動が取れるようにノイズの中から信号を検出します。

Applied Intelligence は、データセットが大規模で複雑な場合でも、明確なガイダンスを提供します。機械は、人間が再現できない規模のデータのパターン、傾向、エラーを特定することに長けています。適切な Applied Intelligence は、テレメトリデータから可能な限り早期に問題を検出し、イベントを関連づけて優先順位を設定することで、ノイズを削減し、疲労を警告します。Applied Intelligence は、問題の真の根本原因とその解決方法を迅速に特定するための推奨事項を含む、関連するコンテキスト、ガイダンス、および提案を用いて、インシデントアラートを自動的に充実させることができます。

実際に使用されている Applied Intelligence の例を以下に示します。チームにアプリケーションの応答時間のしきい値違反に関するアラートが送信されます。人工知能が、アラートの発生する6時間前に、アプリケーションに関連するスループット、レイテンシーエラー、トランザクション信号を自動的に調べます。このシナリオでは、人工知能がアプリケーションに依存するデータストア内のレイテンシーを検出し、データベースの問題とアプリケーションの応答時間の遅れの間の直接的な関係を明らかにします。ここでは二重のメリットがあります：

1. すでに重要なトラブルシューティング分析が実行され、MTTD（平均診断時間）が短縮されているため、チームは根本的な問題をすみやかに解決し、さらにMTTR（平均復旧時間）を短縮できます。

2. 多くのデータを読み込んでトレーニングすることで、Applied Intelligenceがさらに有益になります。軽度のアラームや誤警報からノイズを排除できるため、全体的なアラート疲労を大幅に軽減し、チームが優れたソフトウェアの迅速な出荷に集中できるようになります。



依存関係を可視化し、あらゆるタイプのテレメトリをリアルタイムで掘り下げて調べることができれば、システムの問題を素早く簡単に把握し、問題のトラブルシューティングを行って、データの背後にある「原因」を突き止めることができます。技術環境を自動的に効率良くモデル化すると、キュレートされた視覚化データによって、誰でも簡単に根本原因を特定できるようになります。大規模なデータセットに人工知能を適用することで、データ内の相関関係が明らかになり、担当者がそれぞれの得意分野で力を発揮できるようになります。つまり、厳しい状況において、微妙な判断を下すことができるのです。

CHAPTER 6: プログラマビリティ

可観測性データをビジネスの成果に関連付けるのは、組織が成熟したデジタル企業になる上で避けられない重要なステップです。ビジネスの成功を測る重要な指標から始めて、こうした指標の成功に直接影響する主要業績評価指標 (KPI) を特定する必要があります。レイテンシー (遅延)、エラー、ページロードなどのメトリクスは、アプリケーションのパフォーマンスを把握するには不可欠な要素ですが、顧客体験やビジネス KPI に対するアプリケーションの影響を理解する上では、あまり有益とは言えません。

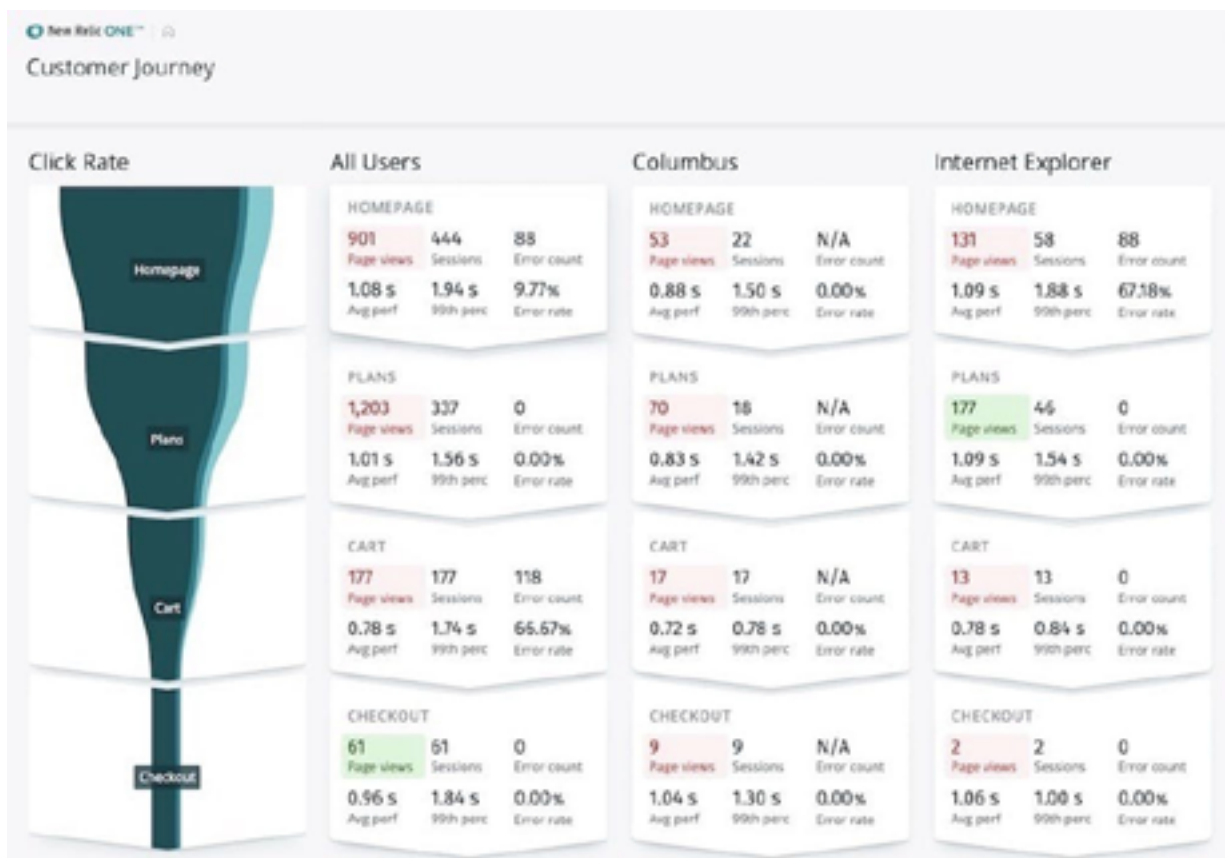
だからこそ、可観測性データをビジネスと関連付け、データに基づいた意思決定を行うために必要なインサイトをチームに提供することが重要です。問題はその方法です。

ほとんどのソリューションの場合、その答えはダッシュボードで KPI を可視化することです。ダッシュボードは、データのアドホックビューを素早く表示するのに最適なツールです。柔軟で強力なツールであり、あらゆる可観測性ソリューションの基盤となるものです。しかし、各企業には特有のテクノロジー環境や独自の KPI が設定されていることを考えると、ダッシュボードにこだわらず、デジタルビジネスに関するデータを取り込み、テレメトリデータと組み合わせるアプリケーションを構築することがこれまで以上に重要になるでしょう。ビジネスのデータを可観測性プラットフォームで関連付けることで、アプリケーションはインタラクティブでキュレートされた体験を提供します。ワークフローが組み込まれていることも多く、リアルタイムで外部データを組み合わせることもできます。こういった操作はダッシュボードではできませんが、アプリケーションなら可能です。

アプリケーションでビジネスデータとテレメトリデータを関連付けるには、可観測性ソリューションはプラットフォームであり、その上で構築できることが条件になります。プログラム可能である必要があります。

独自のニーズに合わせてアプリケーションを構築できる可観測性プラットフォームがあれば、以下のような作業が可能になります。

- ソフトウェアへの投資に優先順位を付け、リアルタイムで投資の効果を測定する。
- 豊富なコンテキストを使用して、テクノロジー、ビジネス、顧客の関係を理解する。
- 特定の KPI に直接影響する意思決定をデータに基づいて行う。
- テクノロジー環境だけでなく、企業特有のビジネスをモデル化するために構築されたインタラクティブで視覚的な表示により、理解を共有する。



プログラム可能な可観測性プラットフォームがチームにあれば、別のツールを導入する必要がなく、一元化されたアプリケーションを構築することができます。緊急時にツール間でコンテキストを切り替える手間を省き、別のシステムのプロビジョニング、運用、保守、監視に要する時間と労力を削減できる、別のツールを購入、構築、学習するコストも削減できるなど、多くのメリットがあります。

CHAPTER 7: あらゆる機能を一元化

ソフトウェアのイノベーションが進むにつれて、世界はますますスピードアップし、複雑化していくでしょう。現在の最新テクノロジーやハイテクのトレンドが数年前には予想もできなかったように、次に何が起こるのかは知る由もありません。ただわかるのは、こうした絶え間ないイノベーションが続き、複雑さが増すことで、素早く行動し、採用するテクノロジーを増やし、光の速さで対応しながらもエラーをゼロに抑えるという過度の期待がチームにかけられています。また、自動化できる作業を増やし、競争を含む他社によって設定された顧客の期待値に継続して応え、最先端の顧客体験を届けなければなりません。

こうした課題を考慮すると、複雑さとリスクを軽減し、コストを抑え、わかりやすく、優れた操作性でスキルギャップ（能力の格差）を埋め、横断的でコンテキストを収集しやすい組織内のどのチームも障壁なく使えるプラットフォームが重要です。チームがテレメトリデータやビジネスデータを一元的に確認でき、素早く意味を引き出して適切な行動を取るために必要なコンテキストを取得するとともに、ユーザーと企業にとって意味のある方法でデータを処理できる。そんな、1台ですべてが処理できるプラットフォームが求められているのです。

可観測性プラットフォームの要件：

- オープンソースと独自のソースからテレメトリデータを収集し、一カ所で組み合わせる。このオープンインストゥルメンテーションでは、ソースに関係なく、あらゆるデータの相互運用が可能になるので、問題や緊急事態が発生した場合でも、ツールが急激に増えることがなく、コンテキストを切り替える手間を軽減します。
- エンティティ間の関連付けと関係性を形成し、これらのつながりを適用して、データをより深く理解できるようにコンテキストと意味を作成します。コンテキストは、最も重要な情報を強調する**キュレートされた**ビューで提示します。
- その上にカスタムアプリケーションを構築する機能を提供します。ダッシュボードとは異なり、アプリケーションはインタラクティブでキュレートされた体験を提供します。ワークフローが組み込まれていることも多く、リアルタイムで外部データを組み合わせることもできます。プログラマビリティは、可観測性の可能性を再定義します。

オープンで、関連付けされたプログラム可能な可観測性プラットフォームがあれば、迅速なイノベーション、デプロイの迅速化、労力の削減、コストの削減、限られた時間と集中力に優先順位を付けるなどビジネスへのメリットは計り知れません。これらのすべての要素は、データ、システム、顧客への理解を深め、共有することにつながります。つまり、企業文化の向上、ビジネスの成長につながります。デジタルサービスのパフォーマンスや顧客とソフトウェアのエンゲージメントをリアルタイムで把握できるので、毎日取り組んでいる最重要事項、つまりビジネスの成果を出すことに集中して取り組むことができます。

New Relicは代表的なデジタルインテリジェンス企業であり、フォーチュン100社の40%以上の企業にフルスタックの可視性と分析機能を提供しています。New Relicのデジタルインテリジェンスプラットフォームは、デジタルビジネスの原動力となる実用的な情報を提供します。あらゆる規模の企業が迅速に問題を解消し、デジタルカスタマーエクスペリエンスを向上させるためにNew Relicを使用してアプリケーションやインフラの監視を行っています。詳細は、<https://newrelic.co.jp/>をご覧ください。